AI 웃음 감지 모델 개발 포트폴리오

양한빈 | AI Engineer

실시간 웃음 감지 AI 모델 설계 및 최적화

`PyTorch` `ONNX` `MobileNetV3` `LSTM` `Fine-tuning`

프로젝트 기간: 26-01-07 ~ 26-02-06 (4W)

---

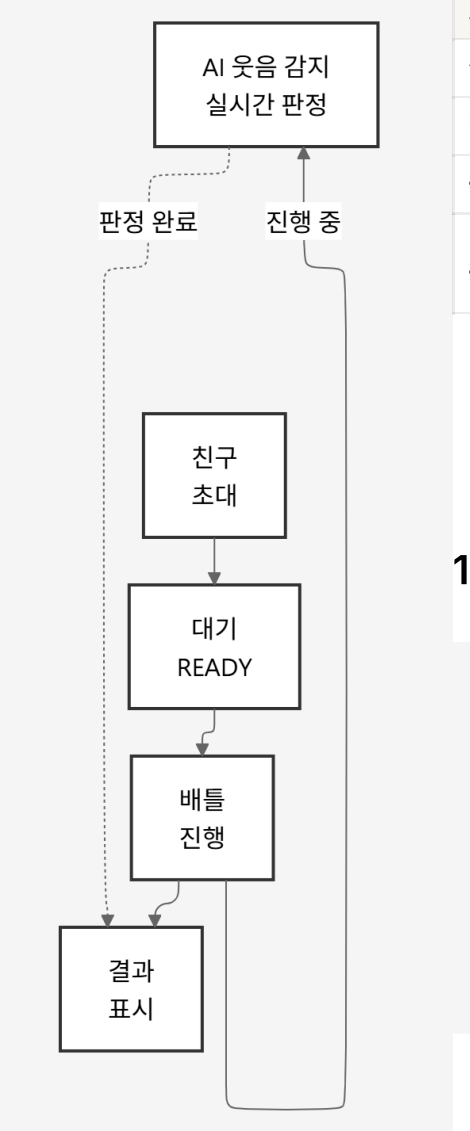
1. 프로젝트 개요

1-1. 서비스 소개: 웃지마게임

🎮 웃지마게임 (ㅇㅈㅁ게임)

"친구와 실시간 화상으로 웃음참기 대결을 즐기는
AI 판정 배틀 플랫폼"

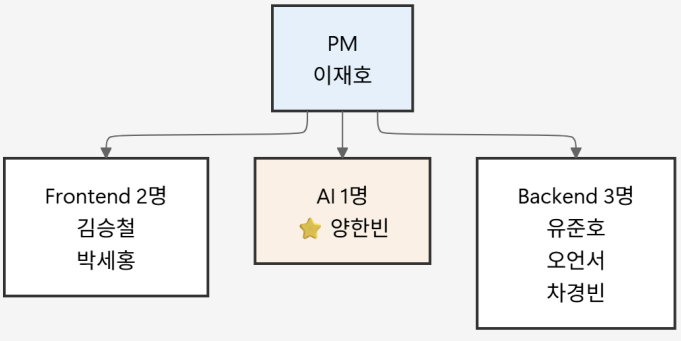
1-2. 게임 플로우



1-3. 핵심 기술 요구사항

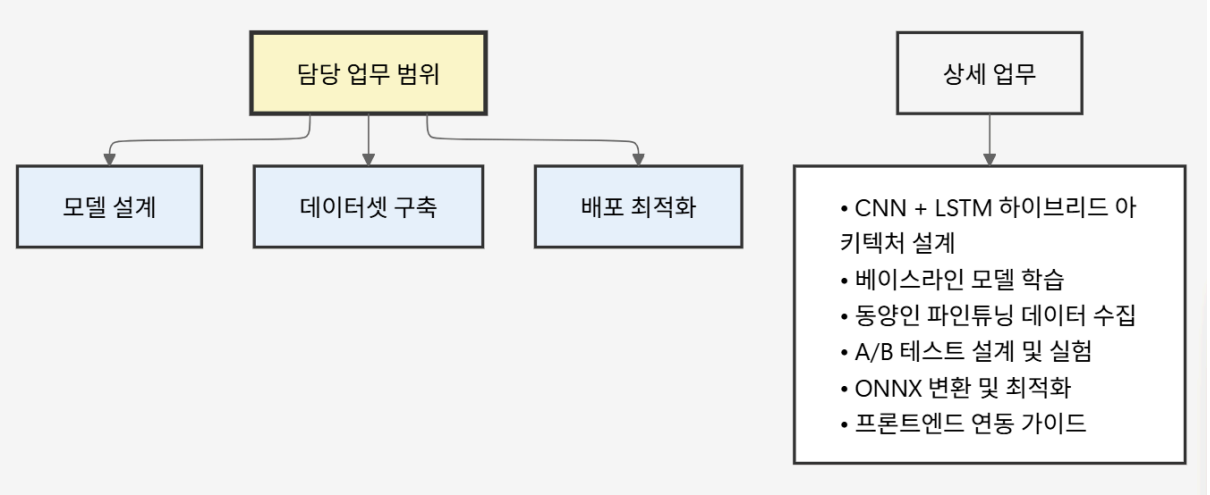
요구사항	목표	난이도
실시간 처리	30 FPS 이상	⭐⭐⭐⭐
낮은 레이턴시	< 50ms	⭐⭐⭐⭐⭐
높은 정확도	85%+ Accuracy	⭐⭐⭐⭐
브라우저 구동	ONNX Runtime Web	⭐⭐⭐

1-4. 팀 구성

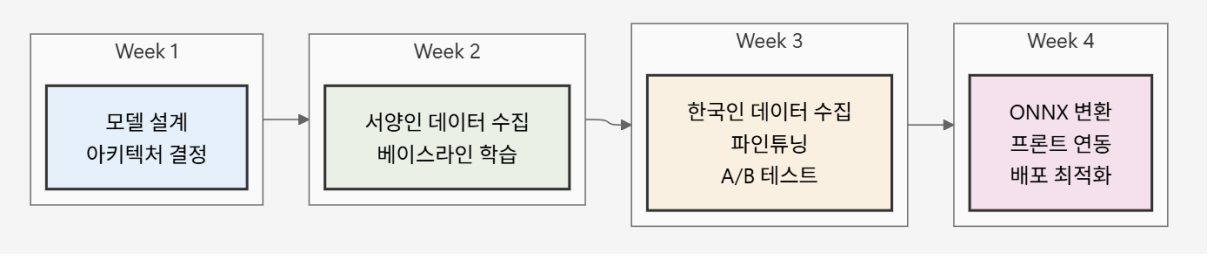


2. 나의 역할 및 담당 업무

2-1. 담당 역할: AI Engineer (단독)



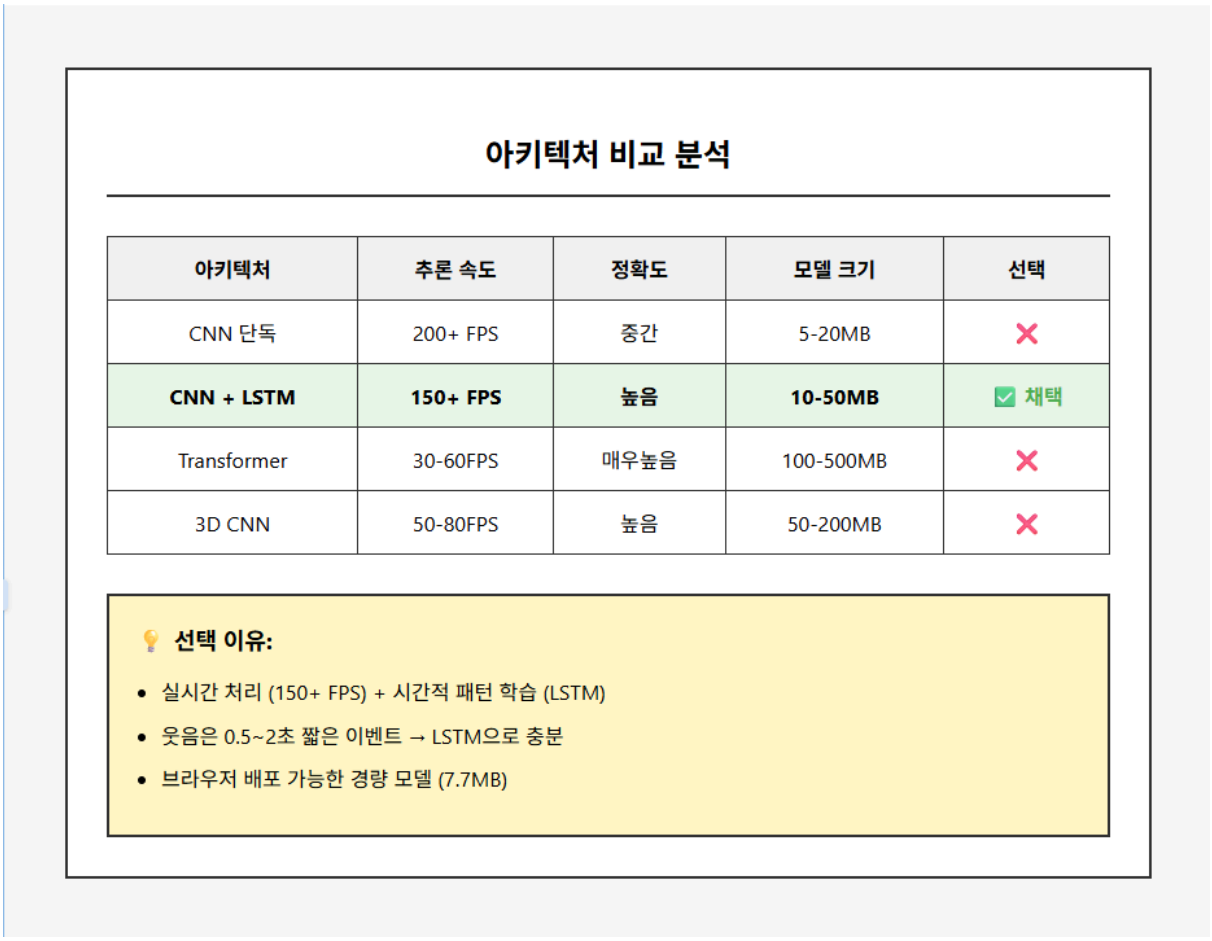
2-2. 업무 타임라인



---

3. 모델 아키텍처 설계

3-1. 왜 CNN + LSTM인가?



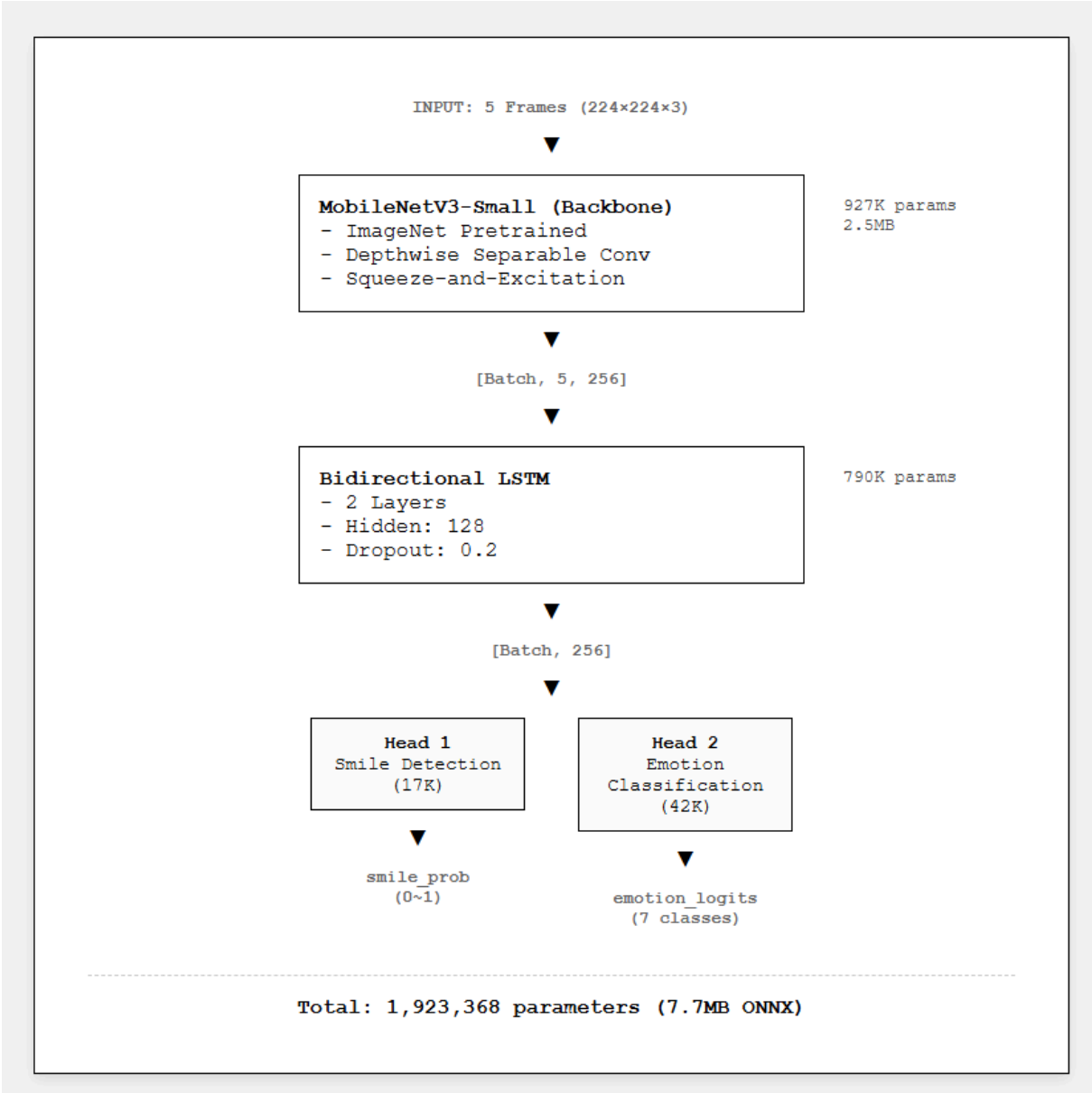
아키텍처 비교 분석

아키텍처	추론 속도	정확도	모델 크기	선택
CNN 단독	200+ FPS	중간	5-20MB	❌
CNN + LSTM	**150+ FPS**	**높음**	**10-50MB**	✅ 채택
Transformer	30-60FPS	매우높음	100-500MB	❌
3D CNN	50-80FPS	높음	50-200MB	❌

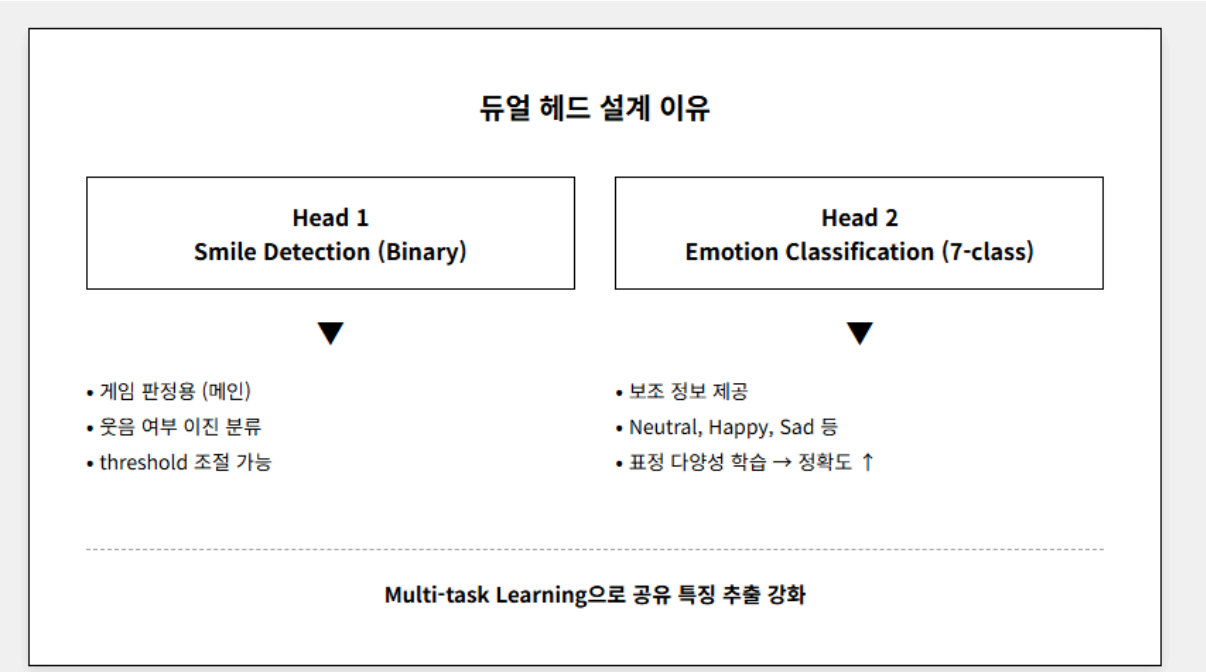
💡 **선택 이유:**

- 실시간 처리 (150+ FPS) + 시간적 패턴 학습 (LSTM)
- 웃음은 0.5~2초 짧은 이벤트 → LSTM으로 충분
- 브라우저 배포 가능한 경량 모델 (7.7MB)

3-2. 모델 아키텍처: DualHeadSmileDetector

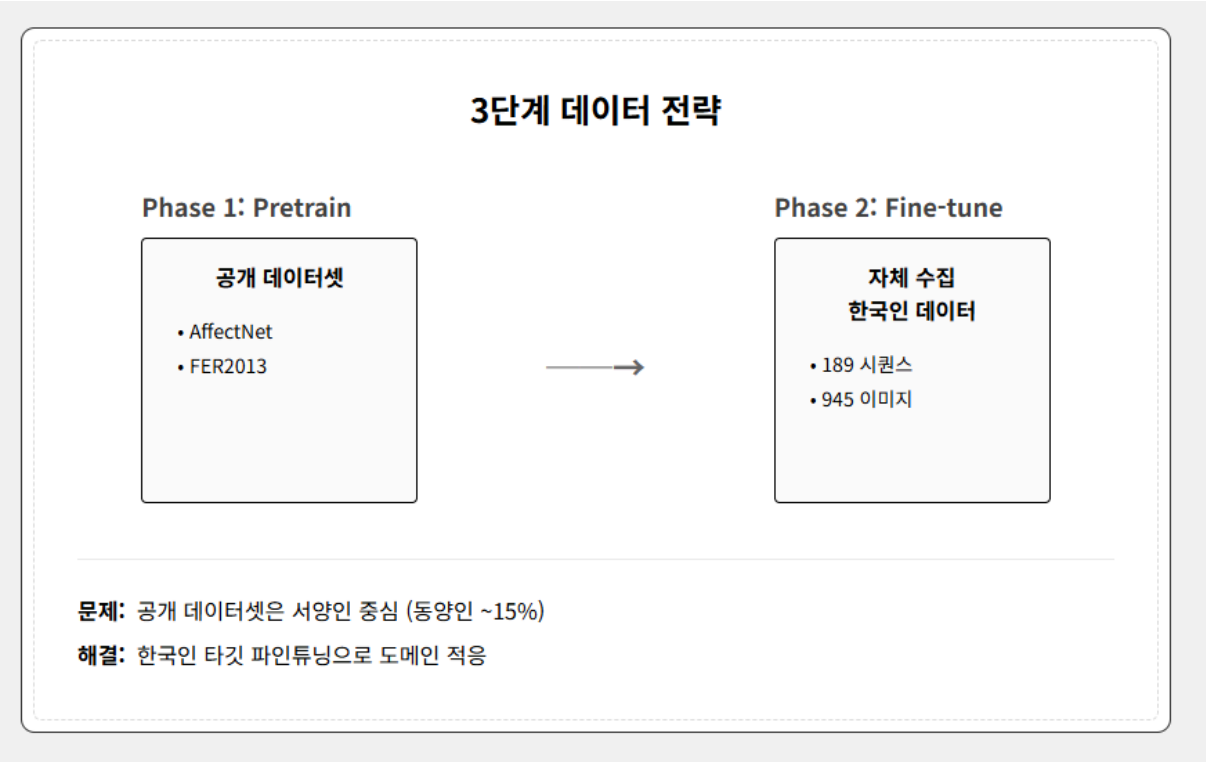


3-3. Y자형 듀얼 헤드 구조의 장점

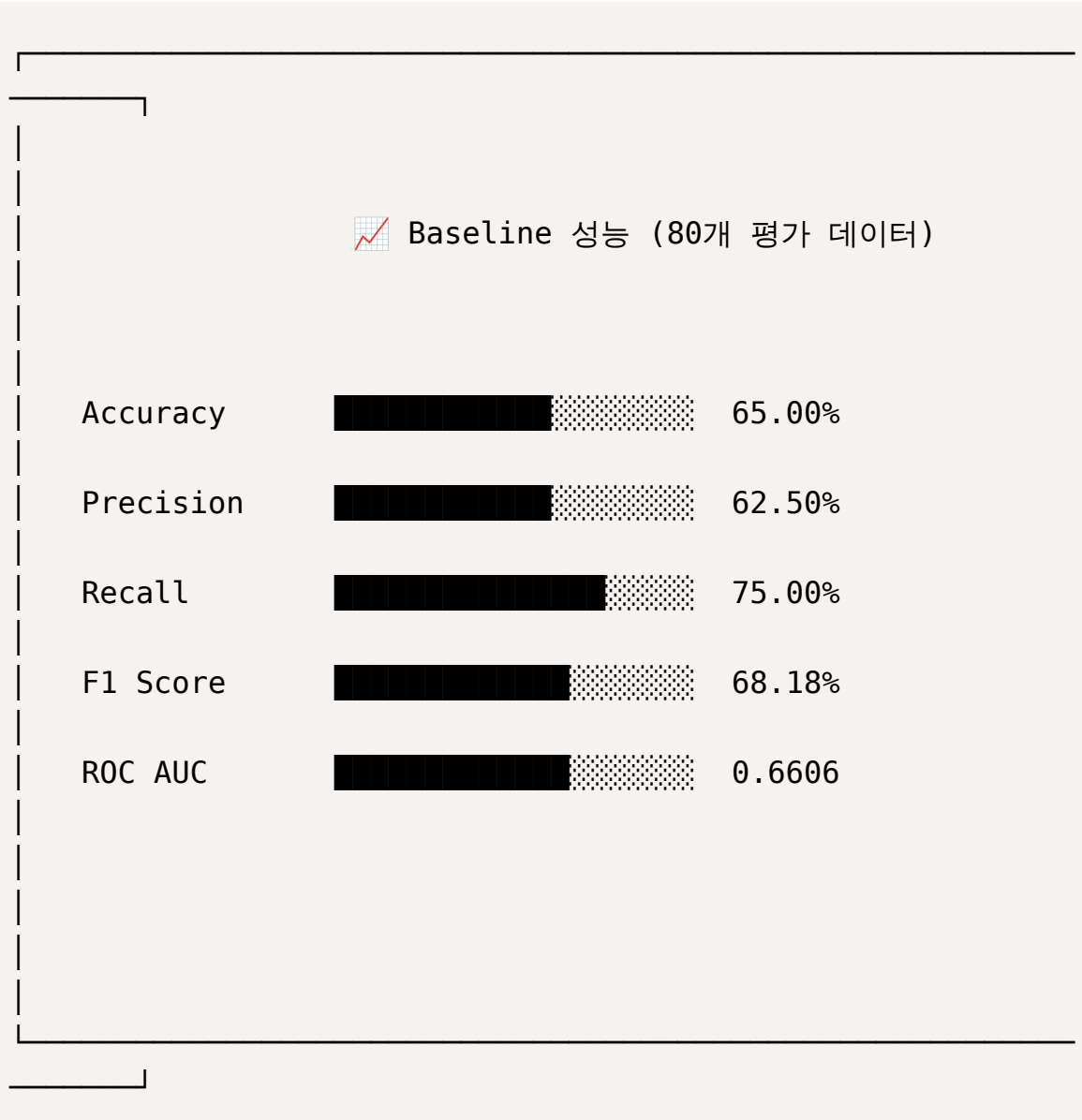


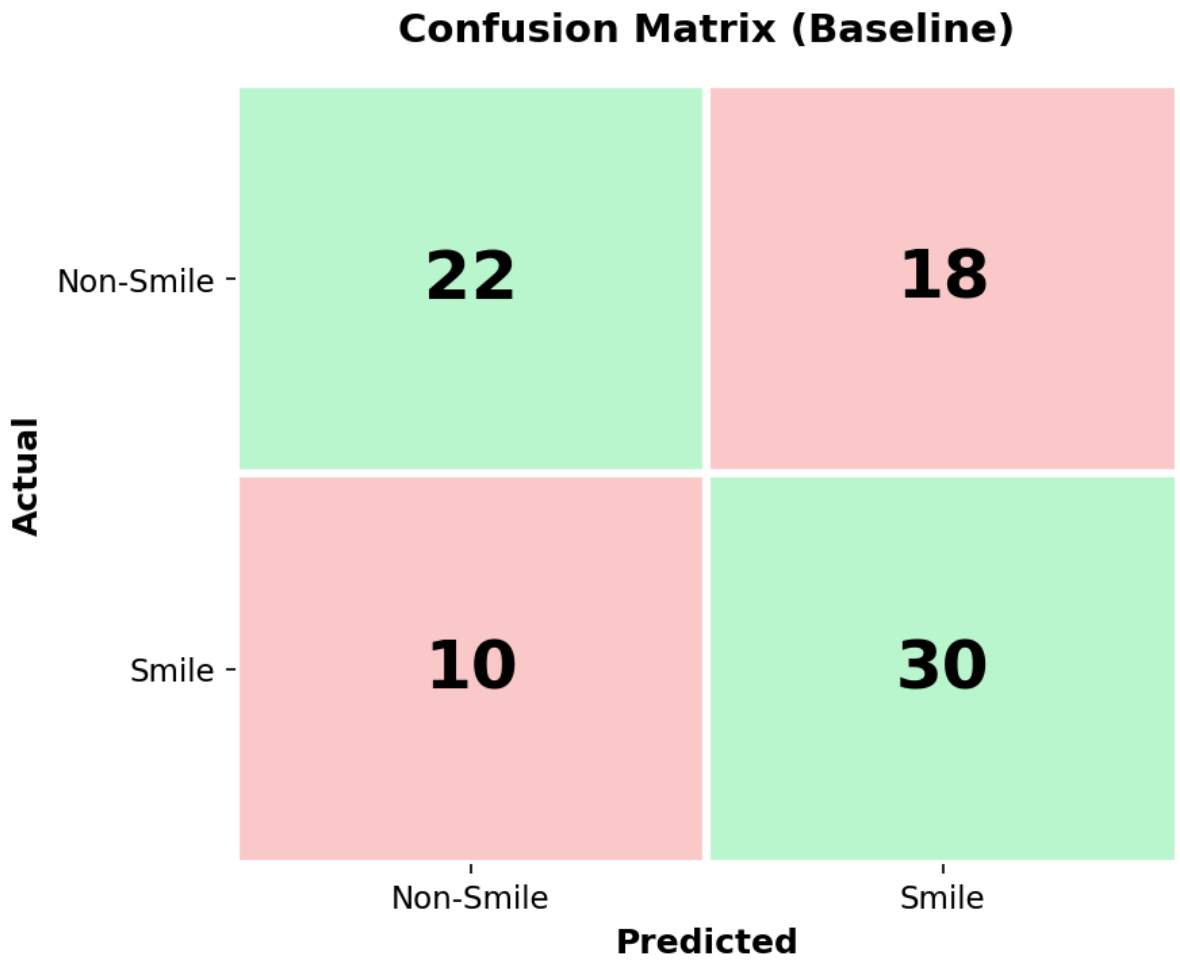
4. 베이스라인 모델 구축

4-1. 데이터셋 전략



4-2. 베이스라인 성능 측정

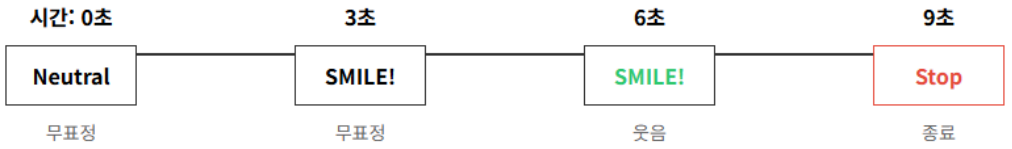

📈 Baseline 성능 (80개 평가 데이터)
Accuracy
65.00%
Precision
62.50%
Recall
75.00%
F1 Score
68.18%
ROC AUC
0.6606



5. 파인튜닝 데이터 수집 전략

5-1. 촬영 프로토콜 설계

9초 학습 시나리오

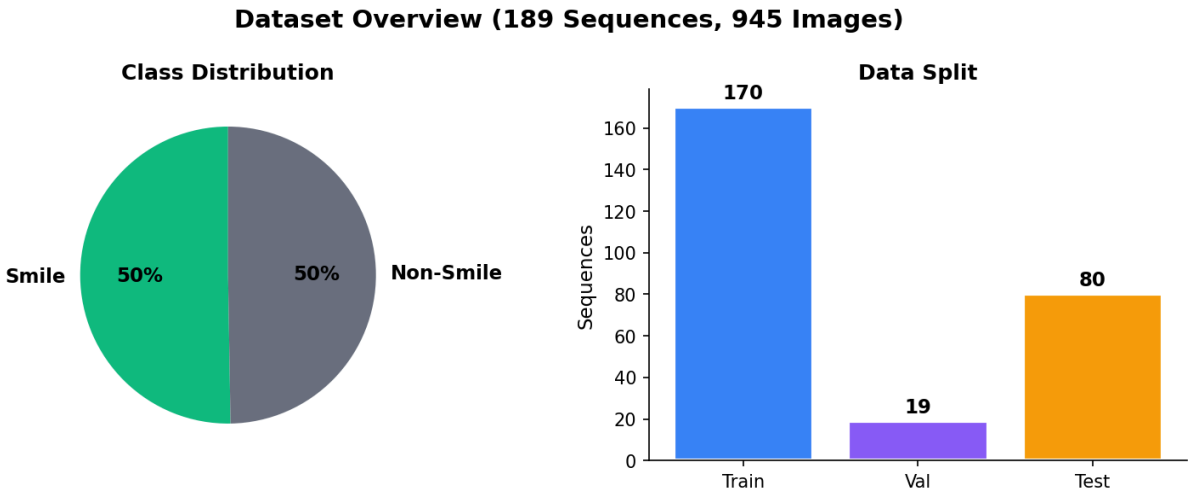


상세 데이터 분석:

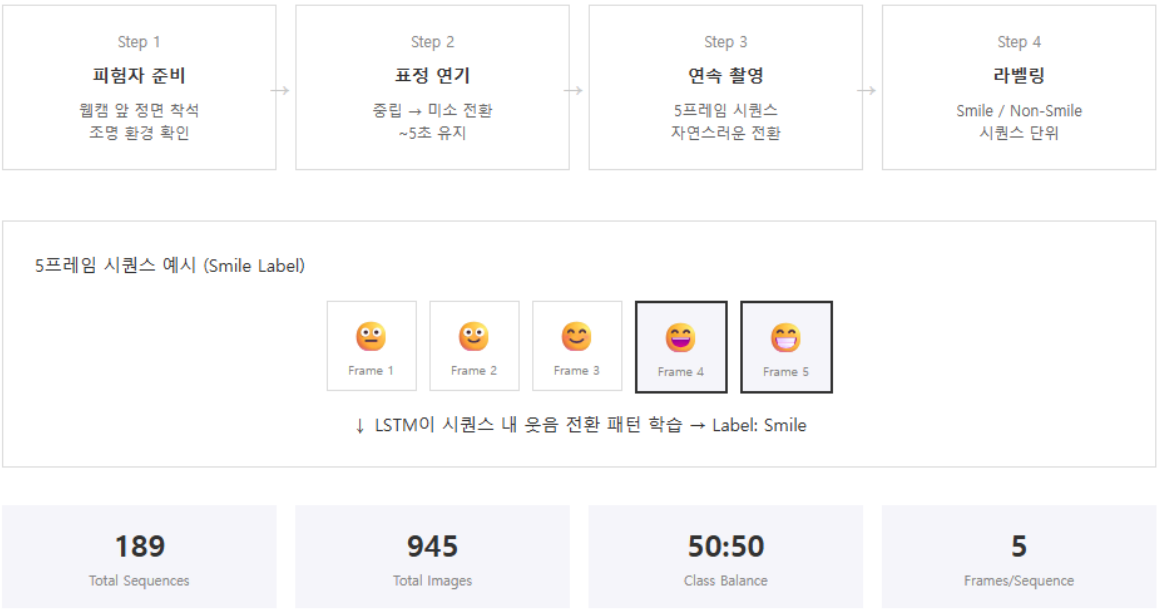
구간	시간	상태	이유
구간 1	1~2초	non_smile	확률적 무표정 유지
구간 2	4~5초	smile	확률적 웃음 감지

💡 **전략:** 애매한 구간(0~1초, 4초) 포함 시 라벨 신뢰성 보장

5-2. 수집 결과



5-3. 시각화 검증 시스템



참고) 데이터셋 구경하러 가기

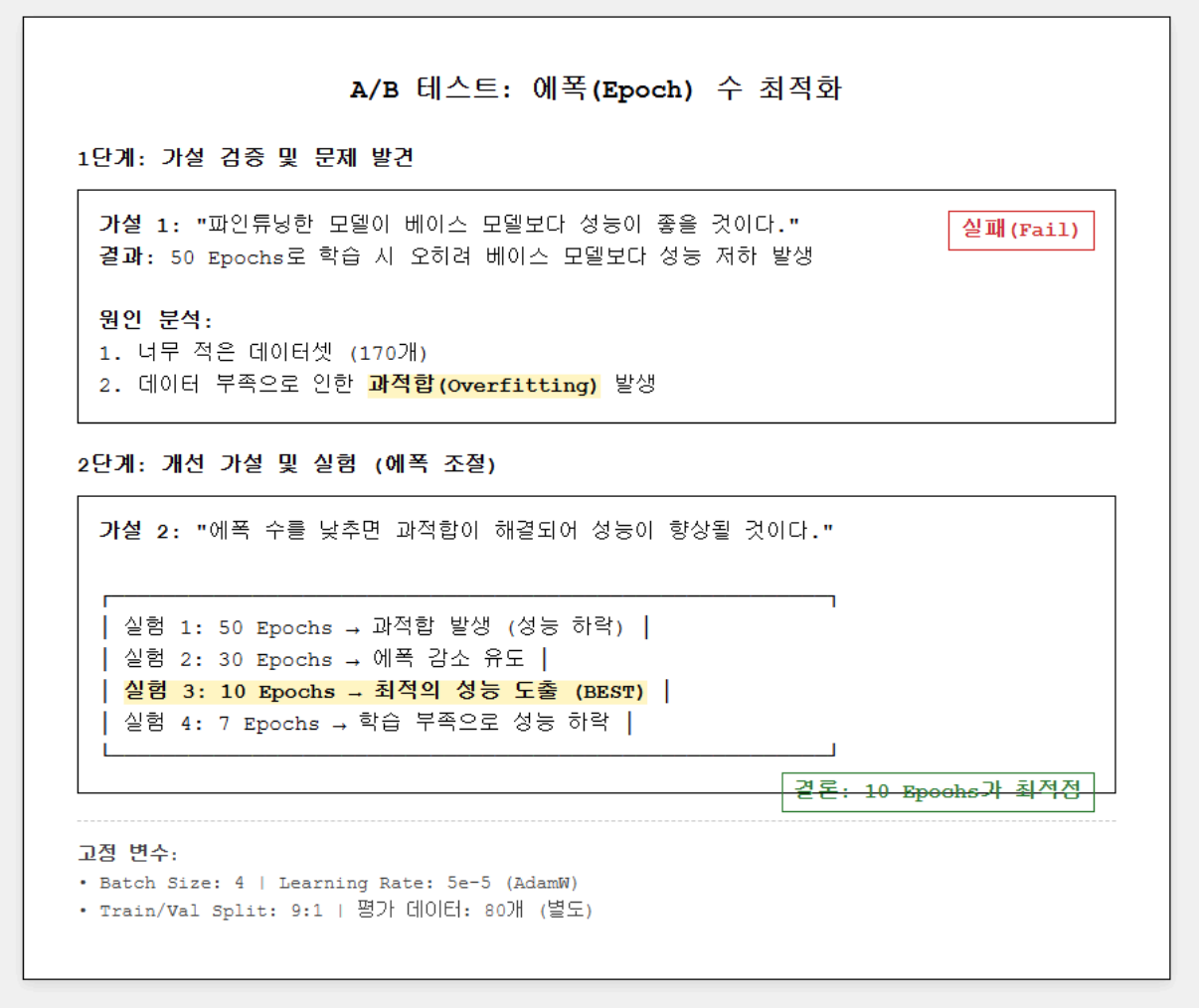
Korean Smile Detection

한국인 웃음 감지 데이터셋 - ssafy

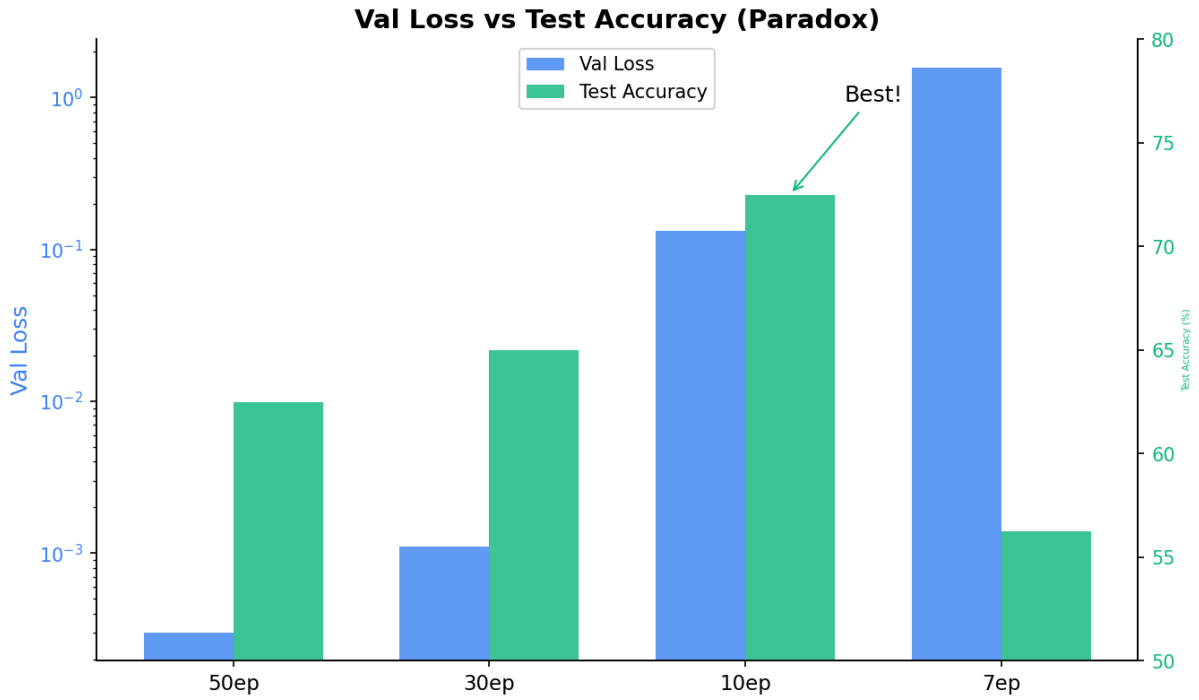
https://www.kaggle.com/datasets/yangonebin/smile-detection-sequences

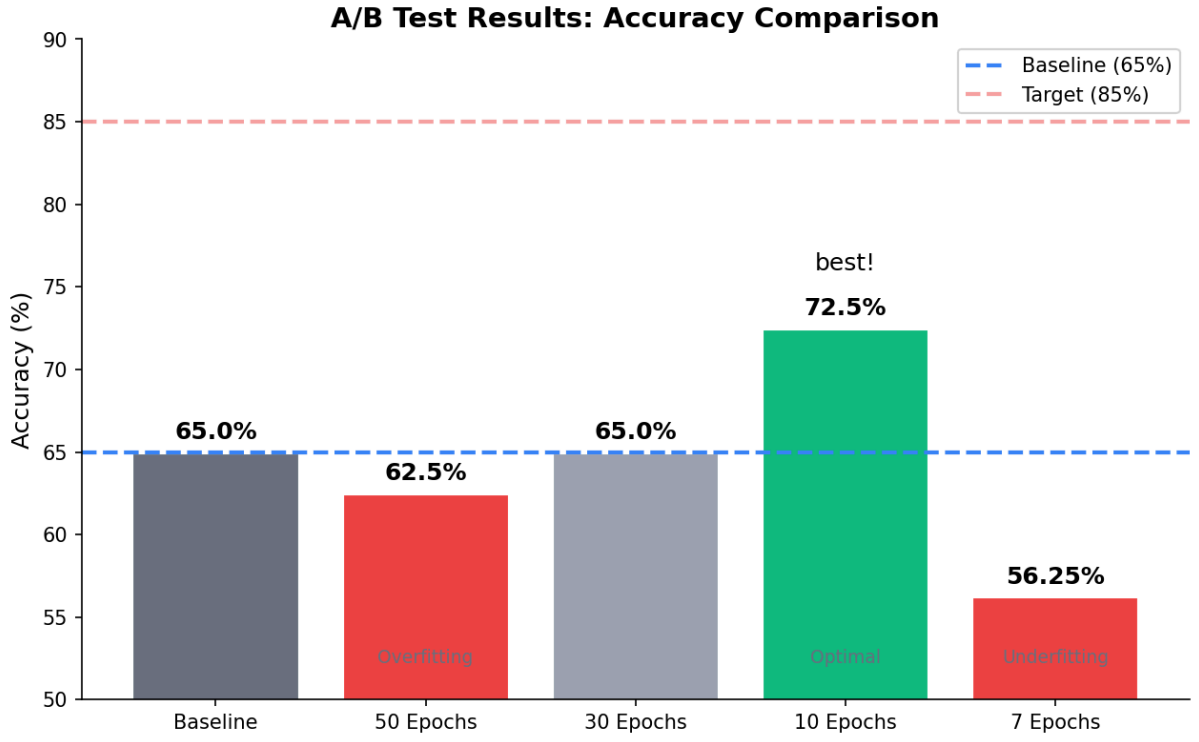
6. A/B 테스트 및 실험 결과

6-1. 실험 설계: 파인튜닝 및 과적합 분석


A/B 테스트: 에폭(Epoch) 수 최적화
1단계: 가설 검증 및 문제 발견
가설 1: "파인튜닝한 모델이 베이스 모델보다 성능이 좋을 것이다."
결과: 50 Epochs로 학습 시 오히려 베이스 모델보다 성능 저하 발생
실패(Fail)
원인 분석:
1. 너무 적은 데이터셋 (170개)
2. 데이터 부족으로 인한 과적합(Overfitting) 발생
2단계: 개선 가설 및 실험 (에폭 조절)
가설 2: "에폭 수를 낮추면 과적합이 해결되어 성능이 향상될 것이다."
실험 1: 50 Epochs → 과적합 발생 (성능 하락)
실험 2: 30 Epochs → 에폭 감소 유도
실험 3: 10 Epochs → 최적의 성능 도출 (BEST)
실험 4: 7 Epochs → 학습 부족으로 성능 하락
결론: 10 Epochs가 최적점
고정 변수:
• Batch Size: 4 | Learning Rate: 5e-5 (AdamW)
• Train/Val Split: 9:1 | 평가 데이터: 80개 (별도)

6-2. 실험 결과 비교

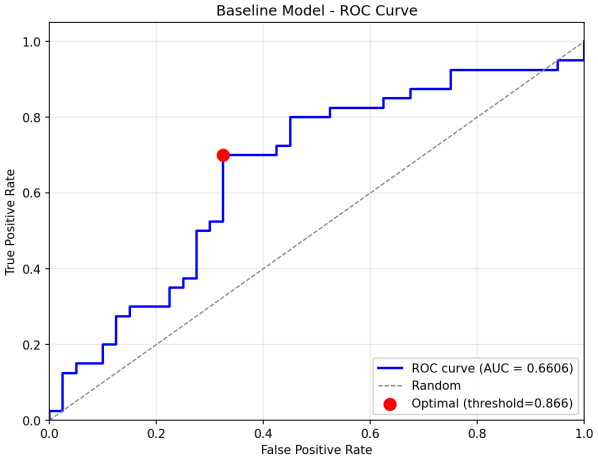


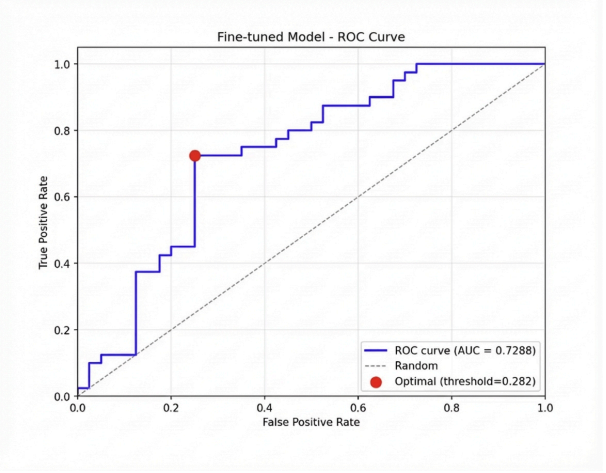


6-3. 최종 결과: 10 Epochs 모델 채택

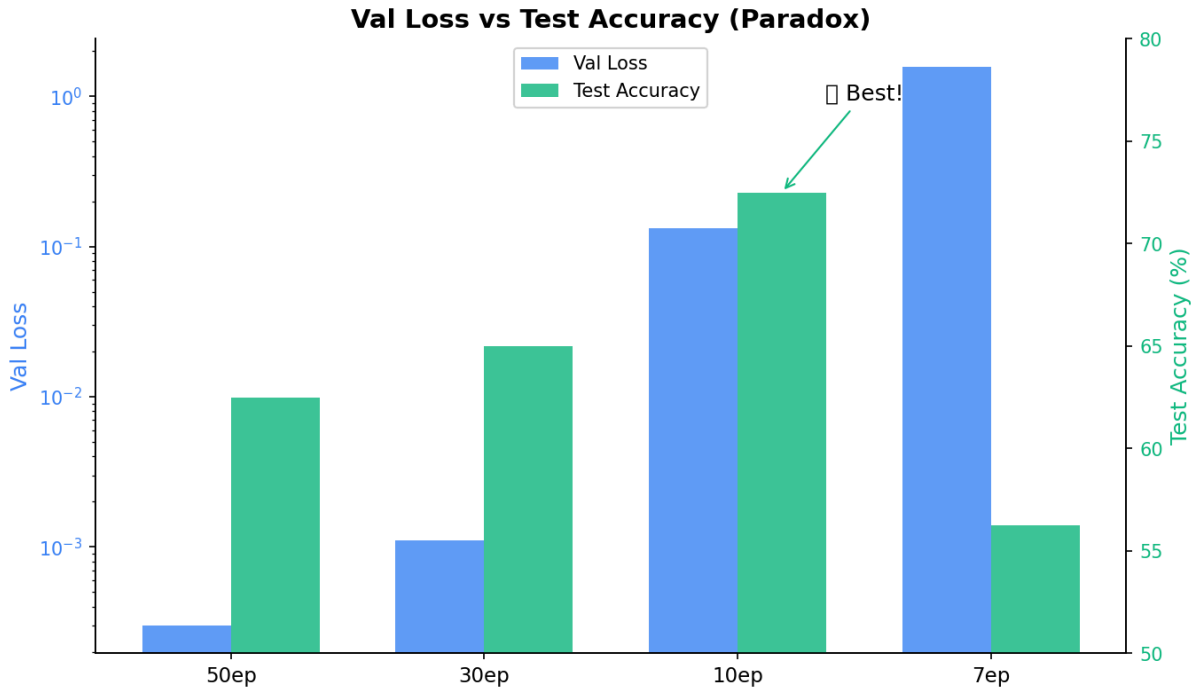
Final Model Comparison: Baseline vs 10 Epochs

Metric	Baseline	10 Epochs	Improvement
Accuracy	65.00%	72.50%	+7.5%p ✓
Precision	62.50%	73.68%	+11.2%p ✓
Recall	75.00%	70.00%	-5.0%p
F1 Score	68.18%	71.79%	+3.6%p ✓
Threshold	0.8659	0.2820	Normalized ✓



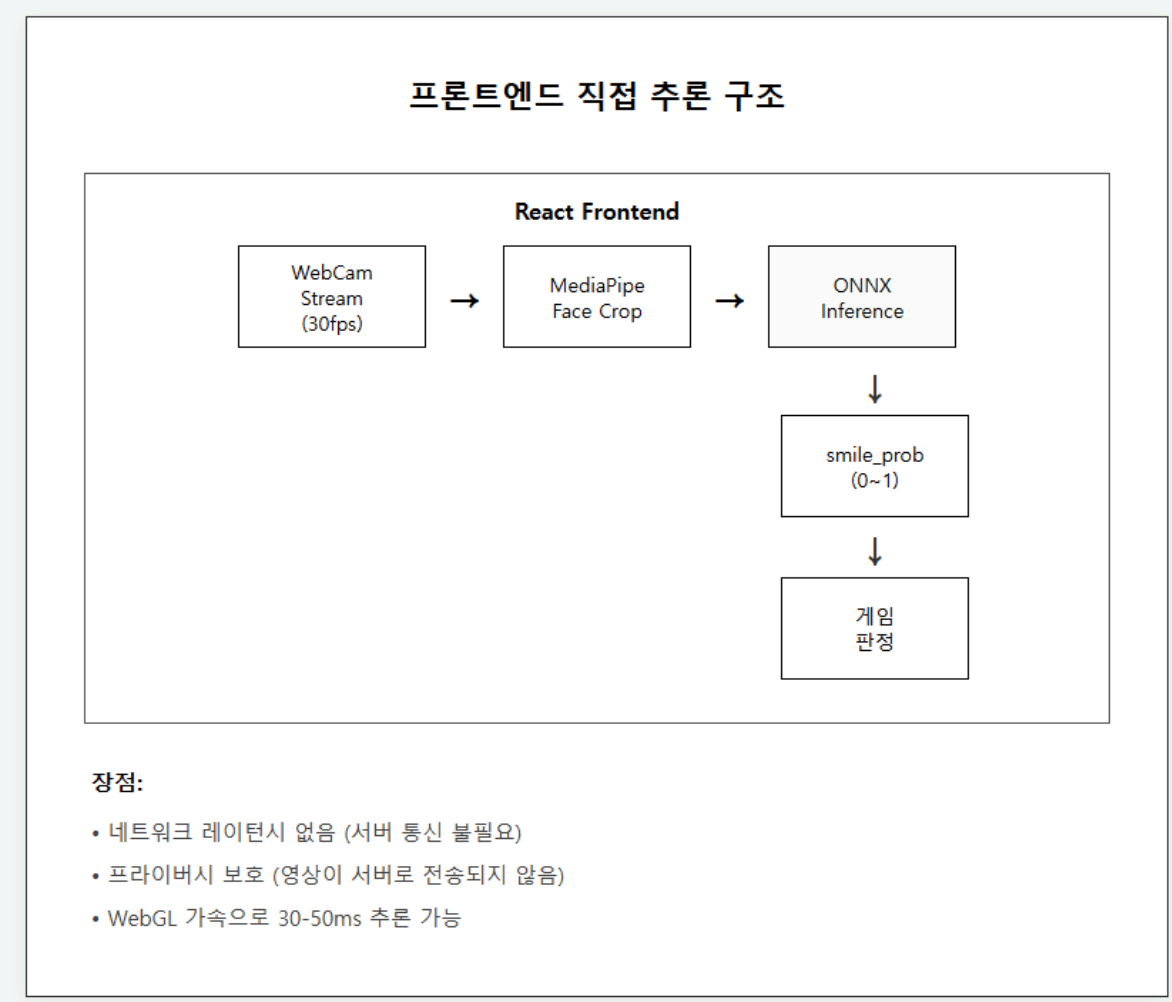


6-4. 에폭 수와 성능의 관계



7. 프론트엔드 연동

7-1. 브라우저 추론 아키텍처



7-2. 프론트팀에 전달한 연동 스펙

7-3. 성능 비교: 서버 vs 브라우저

추론 방식별 성능 비교

방식	레이턴시	장점	단점
서버 API (FastAPI)	~100ms+	GPU 사용 가능	네트워크 지연
ONNX Web (WASM)	60-80ms	네트워크 없음	CPU 사용
ONNX Web (WebGL) ✅	**30-50ms**	**GPU 가속 최고 속도**	**WebGL 지원 필요**

최종 선택: ONNX Runtime Web (WebGL)
→ 30-50ms로 실시간 게임에 적합

8. 기술적 성과 요약

프로젝트 성과 요약

성능 개선

Accuracy: 65% → **72.5%** (+7.5%p)
Precision: 62.5% → **73.7%** (+11.2%p)
F1 Score: 68.2% → **71.8%** (+3.6%p)

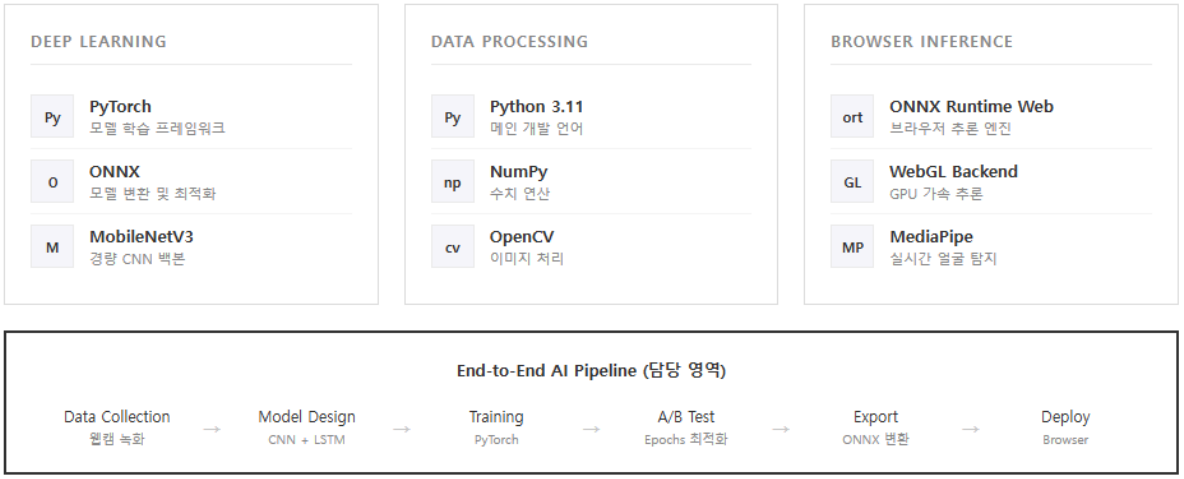
추론 성능

모델 크기: **7.7MB** (브라우저 배포 가능)
추론 속도: **30-50ms** (WebGL)
프레임율: **30 FPS** 이상 달성

실험적 발견

- 적은 데이터에서 과적합이 주요 성능 저하 원인
- Val Loss ↓ ≠ Test 성능 ↑ (역설 발견)
- 10 Epochs가 170개 데이터의 최적점

9. 사용 기술 스택



10. 학습한 점 및 개선 방향

10-1. 파인튜닝의 환상을 깨다

- **파인튜닝은 '만능 열쇠'가 아니다**: 단순히 파인튜닝을 하면 성능이 오르는 게 아니라, 데이터 특성과 모델 용량이 어긋나면 오히려 성능 저하가 날 수 있음을 직접 체감했습니다.
- **데이터 1,000장의 함정**: 1,000장이면 충분한 빅데이터라고 생각했지만, 미세한 표정 변화를 잡아야 하는 정교한 모델에겐 부족하다는 것을 알게 되었습니다. 또한, 양적인 수치보다 중요한 것은 '과적합'을 피하기 위한 최적의 학습 타이밍임을 배웠습니다.

10-2. FastAPI에서 클라이언트 추론으로

- **레이턴시 고민**: 항상 FastAPI로 모델을 배포하던 익숙한 방식에서 벗어나, 0.1초의 반응이 승패를 가르는 게임 환경을 통해 '레이턴시'라는 개념을 처음으로 고려하게 되었습니다.
- **사용자 경험을 최우선한 인프라 설계**: 기술적 관성보다 서비스 목적에 맞는 기술 스택을 선택하는 안목을 길렀습니다.

10-3. 협업으로 일하는 법

- **나의 모델이 아닌 우리의 서비스**: 정확도만 높이는 것에 매몰되지 않고 프론트엔드 팀원이 바로 쓸 수 있는 명확한 '연동 스펙'을 정의하는 것이 프로젝트 완성도의 핵심임을 깨달았습니다.

- **시스템을 통한 협업 관리**: 첫 팀 프로젝트라 배워야할 것이 많았지만, Git 컨벤션을 준수하고 GitLab과 Jira를 통해 일정을 관리하며 협업에서 발생하는 커뮤니케이션 비용을 줄이는 실무적인 프로세스를 경험했습니다.

마무리 멘트

> "실패한 50에폭의 데이터를 숨기지 않고 공유하며 팀원들과 해결책을 찾았던 과정이 가장 기억에 남습니다. 약속된 시스템(Git/Jira) 안에서 효율적으로 소통하며 성장할 수 있었던 소중한 경험이었습니다."

10-4. 향후 개선 방향

Contact

양한빈 | yangonebin@gmail.com

"실험 설계에 의한 모델을 만들겠습니다."